2021 SISBID Dimension Reduction Demo

Genevera I. Allen, Yufeng Liu, Hui Shen, Camille Little

Quick PCA Demo Using College Data

Load in Packages

```
library(ISLR)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.6.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

Load Digits Data

```
#code for digits - ALL
rm(list=ls())
load("UnsupL_SISBID_2021.Rdata")
```

```
data(College)
cdat = College[,2:18]
dim(cdat)
```

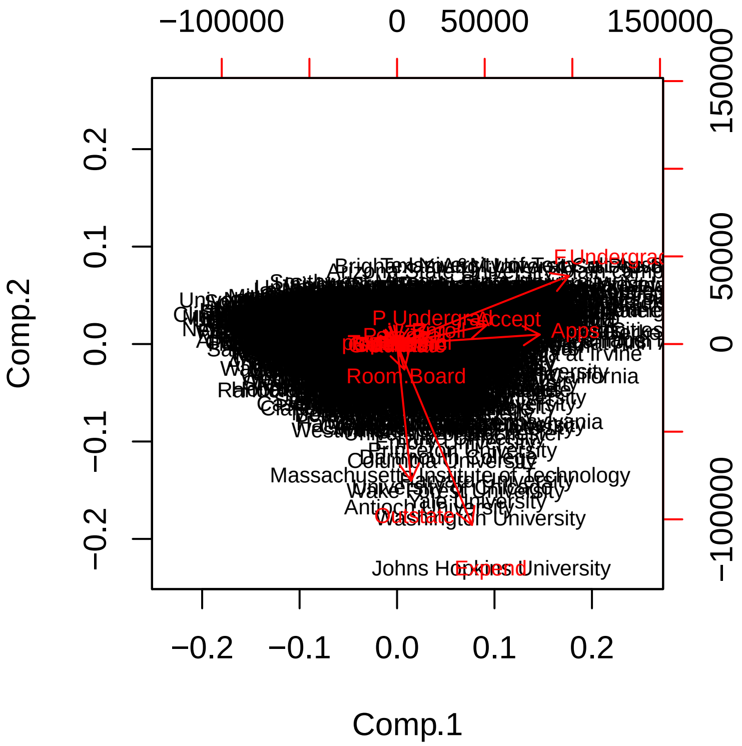
```
## [1] 777  17
```

```
names(cdat)
```

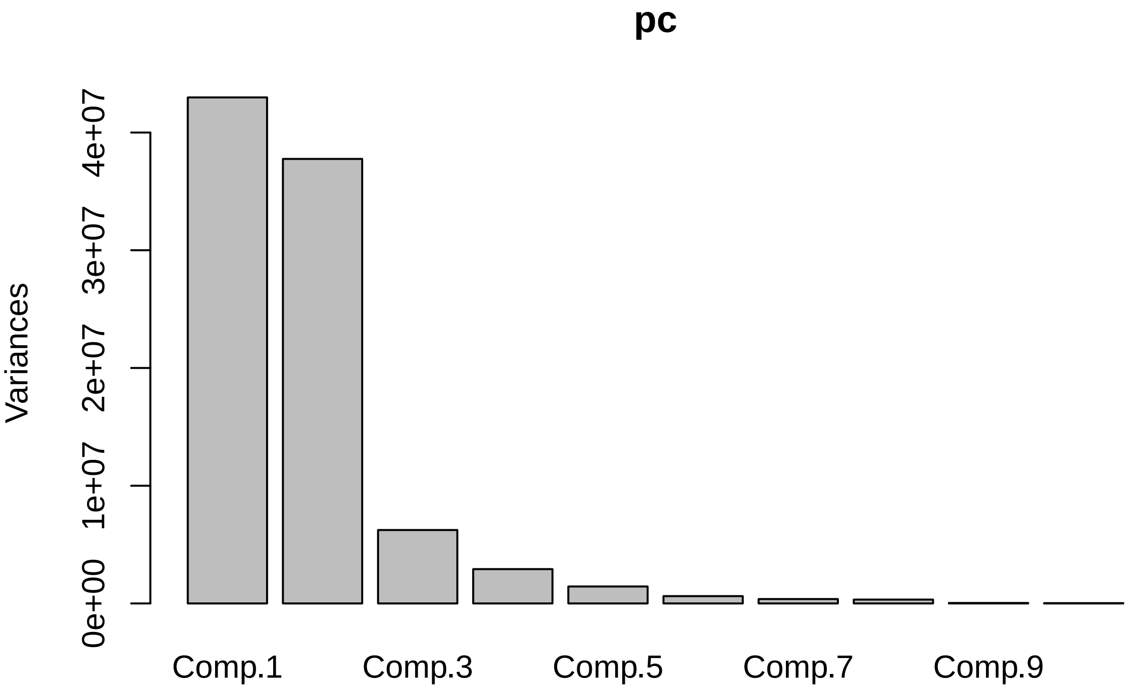
```
##  [1] "Apps"        "Accept"      "Enroll"      "Top10perc"   "Top25perc"
##  [6] "F.Undergrad" "P.Undergrad" "Outstate"    "Room.Board"  "Books"
## [11] "Personal"    "PhD"         "Terminal"    "S.F.Ratio"   "perc.alumni"
## [16] "Expend"      "Grad.Rate"
```

```
pc = princomp(cdat) #default - centers and scales

#Go back and display these plots side by side

biplot(pc,cex=.7)
```

```
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
```

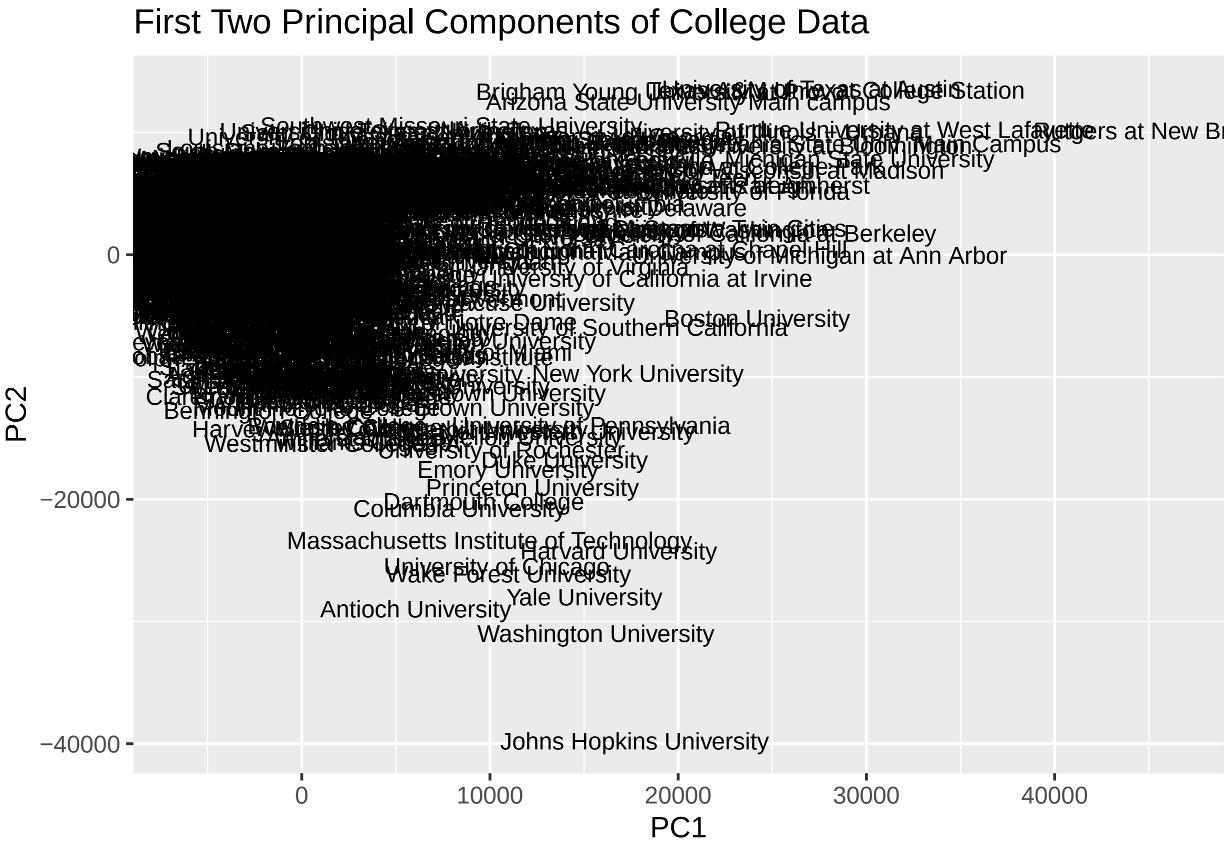


```
screeplot(pc)
```



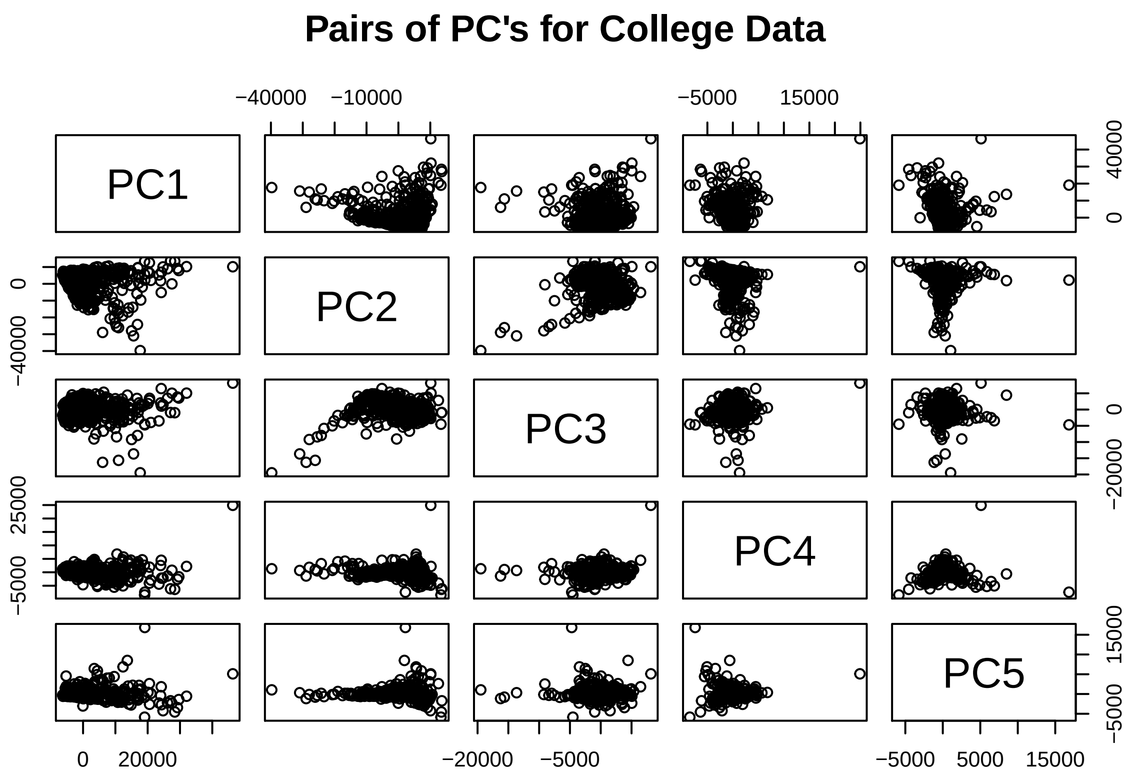
scatter plots - patterns among observations

```
PC1 <- as.matrix(x=pc$scores[,1])
PC2 <- as.matrix(pc$scores[,2])

PC <- data.frame(State = row.names(cdat), PC1, PC2)
ggplot(PC, aes(PC1, PC2)) +
  geom_text(aes(label = State), size = 3) +
  xlab("PC1") +
  ylab("PC2") +
  ggtitle("First Two Principal Components of College Data")
```



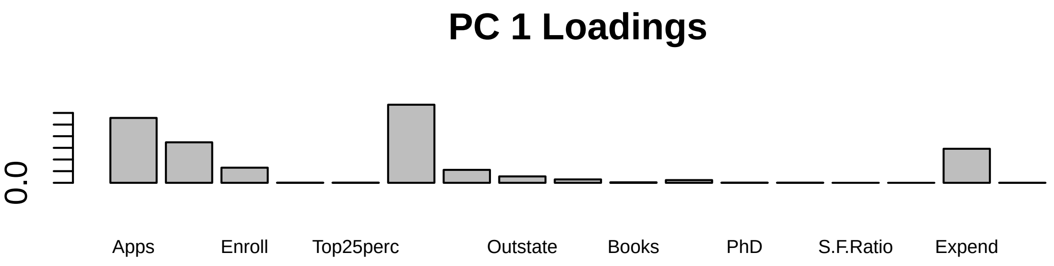
Pairs Plot

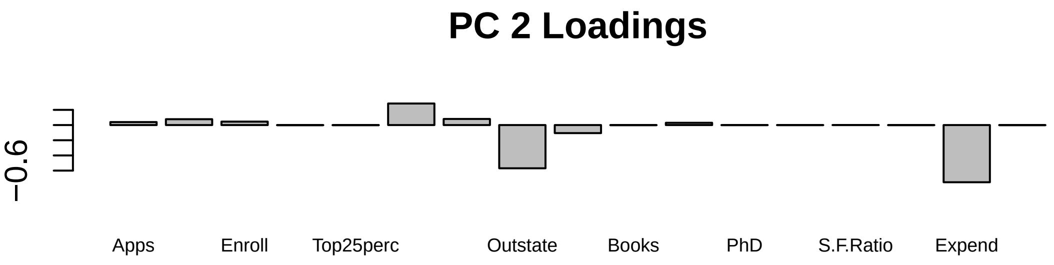
```
comp_labels<-c("PC1","PC2","PC3","PC4", "PC5")
pairs(pc$scores[,1:5], labels = comp_labels, main = "Pairs of PC's for College Data")
```



Loadings - variables that contribute to these patterns

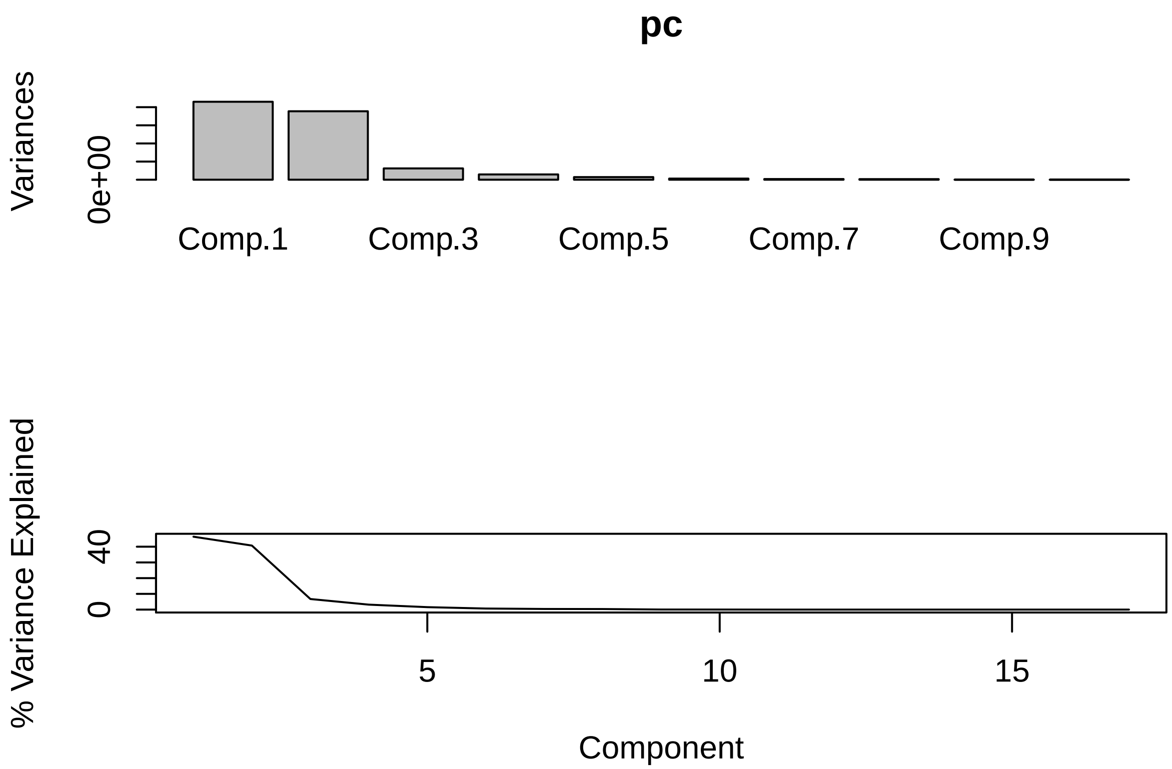
```
par(mfrow=c(2,1))
barplot(pc$loadings[,1],cex.names=.6,main="PC 1 Loadings")
barplot(pc$loadings[,2],cex.names=.6,main="PC 2 Loadings")
```





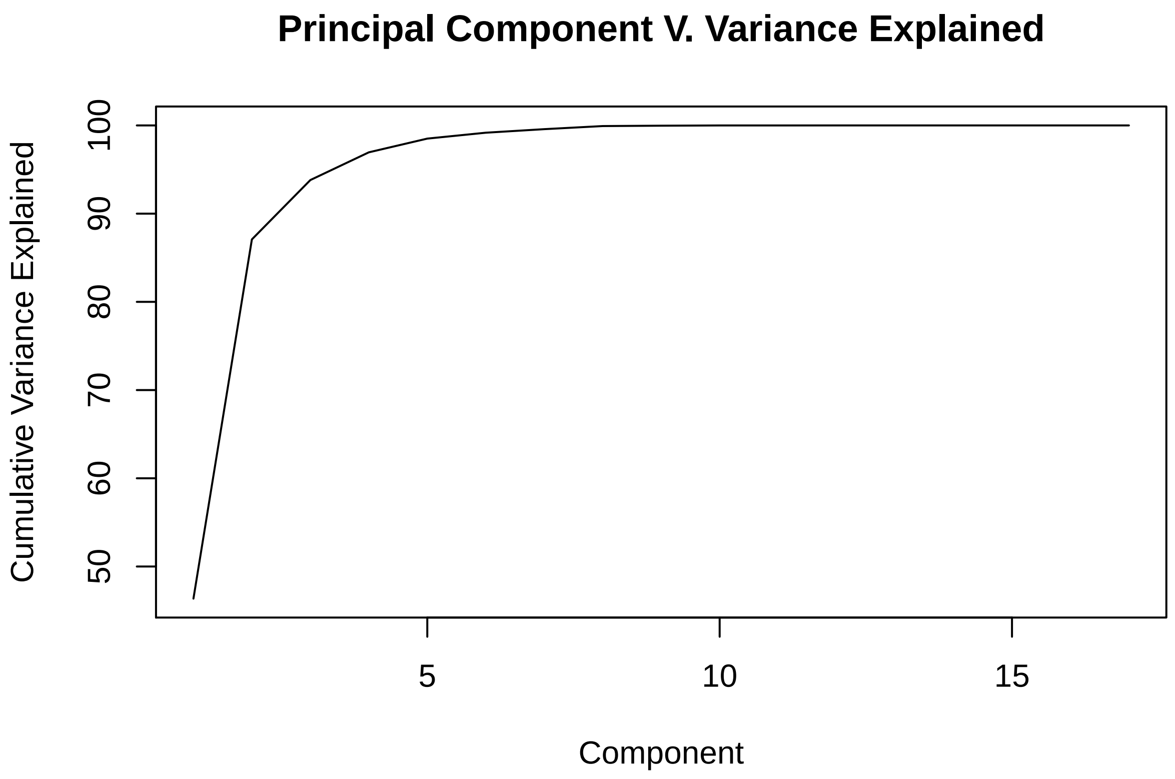
Variance explained

```
varex = 100*pc$sdev^2/sum(pc$sdev^2)
par(mfrow=c(2,1))
screeplot(pc)
plot(varex,type="l",ylab="% Variance Explained",xlab="Component")
```



Cumulative variance explained

```
#cumulative variance explained
cvarex = NULL
for(i in 1:ncol(cdat)){
  cvarex[i] = sum(varex[1:i])
}
plot(cvarex,type="l",ylab="Cumulative Variance Explained",xlab="Component", main = "Principal Component
```



Sparse PCA

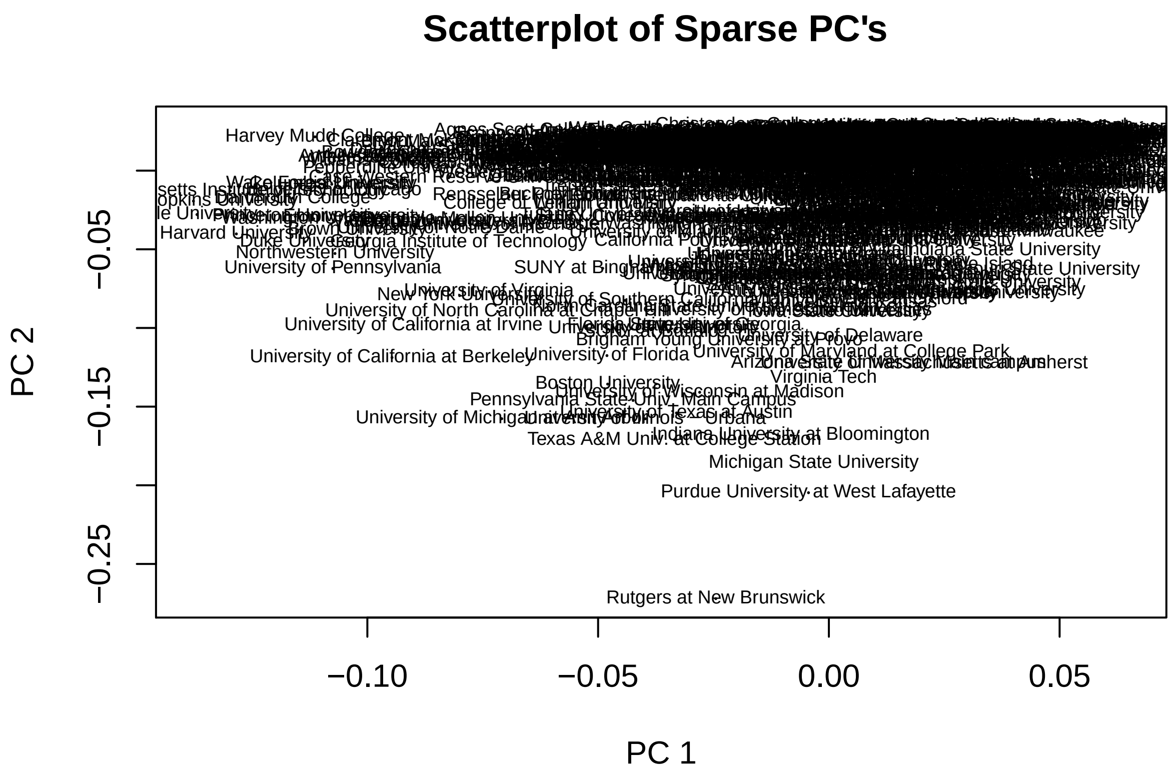
```
library(PMA)

spc = SPC(scale(cdat),sumabsv=2,K=3)
```

```
## 1234567891011121314151617181920
## 1234567891011
## 1234567891011121314151617181920
```

```
spcL = spc$v
rownames(spcL) = names(cdat)
```

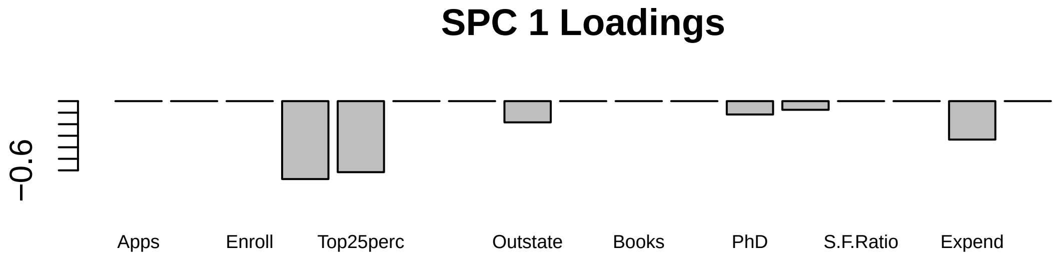
Scatterplots of Sparse PCs

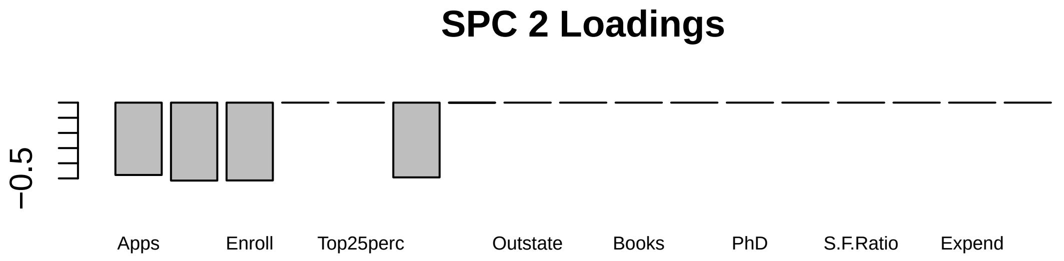
```
i = 1; j = 2;
plot(spc$u[,i],spc$u[,j],pch=16,cex=.2, xlab = "PC 1", ylab = "PC 2", main = "Scatterplot of Sparse PC's
text(spc$u[,i],spc$u[,j],rownames(cdat),cex=.6)
```



Loadings

```
par(mfrow=c(2,1))
barplot(spc$v[,1],names=names(cdat),cex.names=.6,main="SPC 1 Loadings")
barplot(spc$v[,2],names=names(cdat),cex.names=.6,main="SPC 2 Loadings")
```





Try Princomp Function for Digits 3 and 8

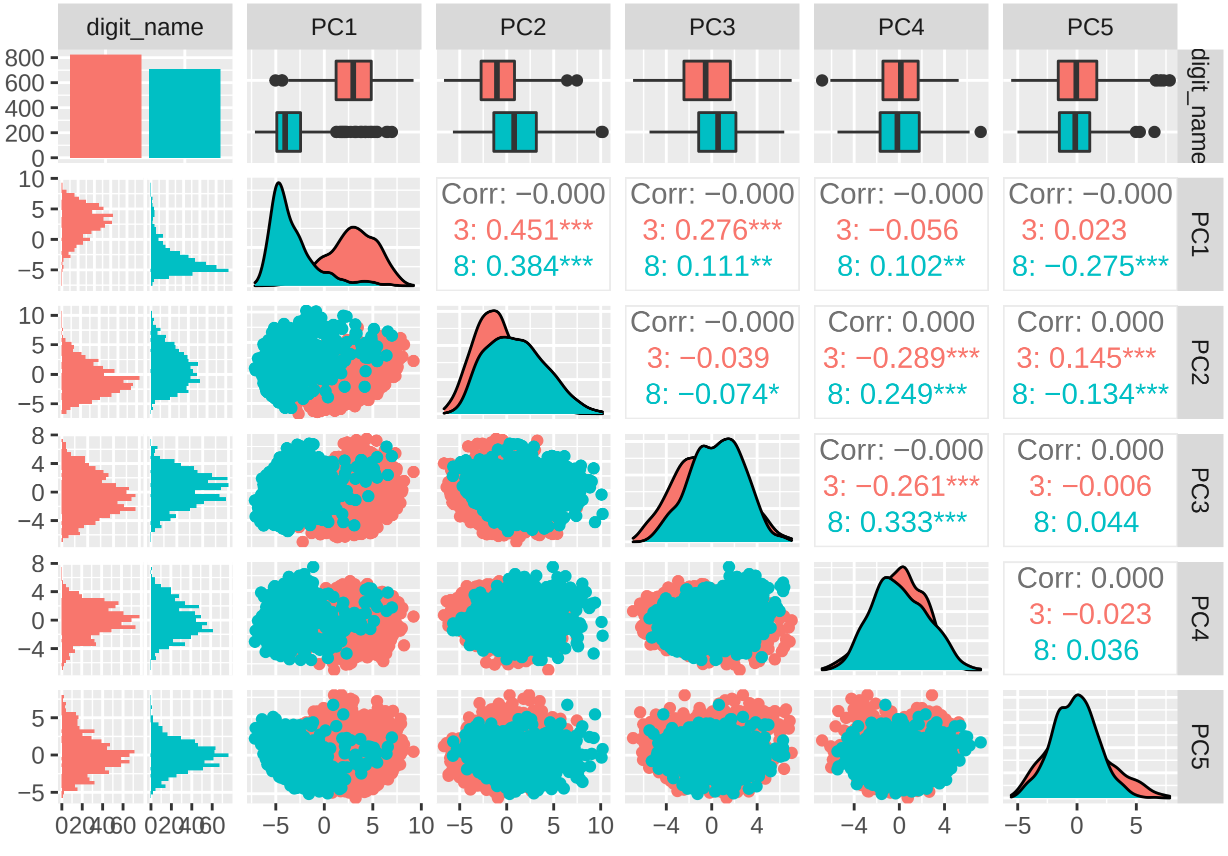
```
dat38 = rbind(digits[which(rownames(digits)==3),],digits[which(rownames(digits)==8),])
```

```
pc = princomp(dat38) #default - centers and scales
```

Pairs plot Using ggpairs

```
PC1 <- as.matrix(x=pc$scores[,1])
PC2 <- as.matrix(pc$scores[,2])
PC3 <- as.matrix(pc$scores[,3])
PC4 <- as.matrix(pc$scores[,4])
PC5<-as.matrix(pc$scores[,5])

pc.df.digits <- data.frame(digit_name = row.names(dat38), PC1, PC2,PC3, PC4, PC5)

ggpairs(pc.df.digits, mapping = aes(color = digit_name))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



PC Loadings

```
par(mfrow=c(3,5),mar=c(.1,.1,.1,.1))
for(i in 1:15){
  imagedigit(pc$loadings[,i])
}
```